

Constraint-based Shaping of Gestural Performance

Guerino Mazzola and Stefan Müller

MultiMedia Laboratory (MML)
Computer Science Department
University of Zürich

guerino@mazzola.ch, s.mueller@computer.org

Abstract

Our ongoing research in performance theory integrates methods for complex instrument parameter spaces and models for musical gestures. The latter are modelled as parametric curves residing in high-dimensional symbolic and physical gesture spaces. This article shortly exposes basic concepts of those spaces and the construction of symbolic gesture curves. It then discusses the problem of fitting physical gesture curves (which are based on their symbolic counterparts) as a function of anatomical and Newtonian constraints. Our solution makes use of Sturm's zero theorem for cubic splines. The resulting curves can be applied to animation of avatar parts in an animation system. This theory is implemented in the latest version of the performance component of a well-known modular software for analysis, composition, and performance.

Keywords: Gestural Performance, Performance Interfaces, Performance Theory, Computer Animation.

1 Introduction

The RUBATO music workstation, first presented by Mazzola and Zahorka (1994), contains a software module called the *PerformanceRubette*, which is able to perform musical scores based on a number of user-controlled analyses (e.g. metric, harmonic, motivic) on them. A performance is calculated in terms of a performance transformation φ from a symbolic score space S to a corresponding physical space P . The score space S is generally built of six note parameters: onset E , pitch H , loudness L , duration D , glissando G and crescendo C . The output of the *PerformanceRubette* is typically a MIDI file containing the resulting performance. The MIDI file can then be played on a built-in MIDI synthesiser or on an external MIDI device. However, we realised that a performance transformation based on the parameters given above is too restrictive for a more realistic

performance; particularly with respect to sound quality. For example, in the case of the violin, it is impossible to specify plucked versus bowed notes with above parameters. Furthermore, the whole area of musical gestures was omitted in the original *PerformanceRubette*. Therefore recent and ongoing research in performance theory at the MML focuses on how note parameter spaces can be extended and how concepts for musical gestures can be incorporated. This includes direct sound synthesis from given physical parameters, and visual representation of musical gestures by means of performing avatars or avatar parts.

Consequently, the model of performance transformations has been extended with spaces lifted to gestures and a corresponding *gesture transformation*, as introduced in (Mazzola, 2002a). Those gesture spaces contain *parametric gesture curves*, which represent musical gestures: the symbolic gesture curve describes the movements of an abstract, *symbolic performer* playing a symbolic instrument. In contrast, the physical gesture curve describes the movements of a *virtual performer* who resides in a geometric 3D space and thus is intended to behave like a *real performer*, playing a real instrument. In addition, physical gesture curves can be used for sound synthesis of physically modelled instruments, for instance by using the kinematics of the performer's movements as sound synthesis parameters. Such methods also support the integration of novel performance interfaces.

This article deals with the construction and the constraint-based shaping of physical gesture curves from a given performed score: we will take off with algorithmically constructed symbolic gesture curves (the construction of which has been dealt with in detail in (Müller, 2003)). Since those curves can contain anatomically impossible fingerings, they first have to be fitted with global anatomical constraints. Then they will be shaped to dynamic constraints, based on Newton's equation, in order to make the movements more realistic. Observe that we shall focus on piano-like instruments throughout the article, however, the theory can be applied *mutatis mutandis* to other instruments.

In the next section, we will cover relevant related work in the area and summarise the basic concepts of gesture spaces and curves. Then, we will focus on the constraint-based fitting and shaping of physical gesture curves. The results section will present examples calculated with the PerformanceRubette. The article concludes with examples and an outlook.

2 Related Work

Throughout this article, we shall not deal with performance theory in general, but will provide basic concepts, where necessary. Refer to (Mazzola, 2002b) for details. An in-depth coverage of ongoing research of RUBATO with respect to computer-aided musical performance and the new PerformanceRubette is given in (Müller, 2002).

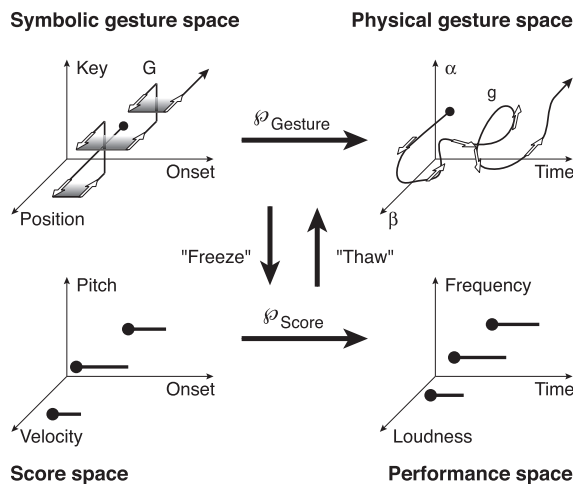


Figure 1. Gesture spaces containing the symbolic gesture curve G and the physical gesture curve g , their inter-relationships, and the relationship to the corresponding instrument spaces.

In computer music and music informatics, musical gestures have mainly been used in the domain of controlling musical instruments. Refer to (Wanderley and Battier, 2000) for an extensive investigation on this subject; in particular, (Wanderley and Cadoz, 2000) discusses several definitions of the term ‘gesture’ in this context. For a coverage of semiotic aspects of music and gestures refer to (Mazzola, 1999) or (Henrotte, 1992).

When attempting to synthesise musical gestures in terms of a geometrical representation, it becomes necessary to have a look at the field of computer animation and computer graphics. Cassell et al. (2001)

have presented a toolkit which can generate appropriate and synchronised non-verbal behaviours and synthesised speech from given input typed text. The output then can be sent to an animation system, which for example renders a *talking avatar*. Refer to (Badler et al., 1993) for a broad discussion of animation of humans, and to (Faloutsos et al., 2001) for recent developments.

3 Gesture Spaces and Gesture Curves

This section summarises the basic concepts of gesture spaces and gesture curves since they will be necessary for understanding the shaping of gesture curves in the following section. For a full account of the theory refer to (Müller, 2003).

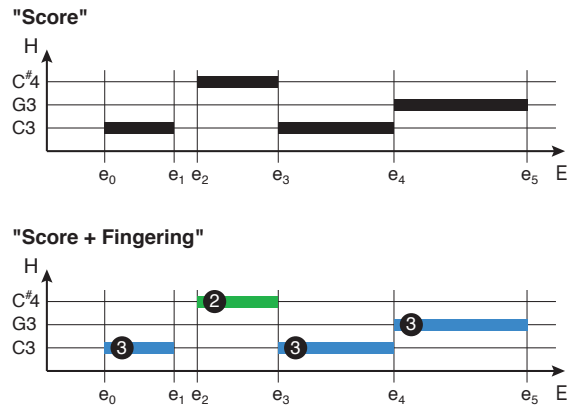


Figure 2. A symbolic score, above without, below with fingering. Horizontal axis denotes onset time, vertical axis pitch.

Performance theory, as implemented in the PerformanceRubette, defines a score space S , a performance space P and a transformation $\varphi_{Score} : S \rightarrow P$ between the two. This situation is shown in Figure 1: the score space and the performance space contain note and tone events, starting at the position of the black dots and having the duration of the length of the line. This model has been extended to *lifted gesture spaces* (Müller, 2002), more precisely a symbolic gesture space and a physical gesture space respectively, together with a *gesture transformation* $\varphi_{Gesture}$. The vertical relationships between the spaces are defined by “freezing” the gesture spaces, or by “thawing” the score spaces. From this point of view, symbols in a musical score (e.g. the notes) can be seen as *frozen gestures*. This proposition is supported by the observation that today’s music notation originated from *neumes*. Neumes are an early form of music

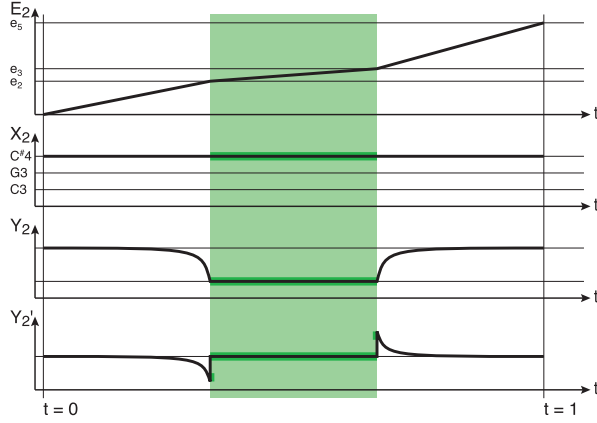


Figure 3. Symbolic gesture curve for finger 2, with curve parameter t running from 0 to 1 on the horizontal axis.

notation (Parrish, 1957), and the word “neume” is actually the Greek word for “hint”. Symbolic music notation can therefore be seen as a highly abstract way of writing down gestures.

In the model of Figure 1, gestures are represented by high-dimensional parametric curves in a particular gesture space. For example, the symbolic gesture curve G in Figure 1 represents a (monophonic) gesture for a keyboard-like instrument, closely modelled after MIDI concepts (“Note on”, “Note off” and “Velocity”, the derivative of the figure’s position coordinate).

The physical gesture curve $g = \wp_{Gesture}(G)$ represents the transformed symbolic gesture curve G . Here, the parameters will be typically be of geometric nature, such as angles between finger segments, and motion parameters, such as velocity and acceleration vectors of the finger tips or of the ankles. Those parameters are represented by the α and β axes in the physical gesture space of Figure 1.

3.1 Construction of Symbolic Gesture Curves

For the discussion of the ‘thawing’ operation, consider the piano-roll like scores in Figure 2. Pitch is given by the vertical axis, onset time by the horizontal axis. The four events reside inside the onset boundaries e_0 to e_5 . The lower score has been annotated with fingering information, which we assume to be given (e.g. manually).

One of the main problems with symbolic gesture curves is the issue that fingers have to move at infinite speed in some cases: for instance at e_3 the second event for finger 2 ends and at the same time finger 3 has to start playing the third event. This problem has

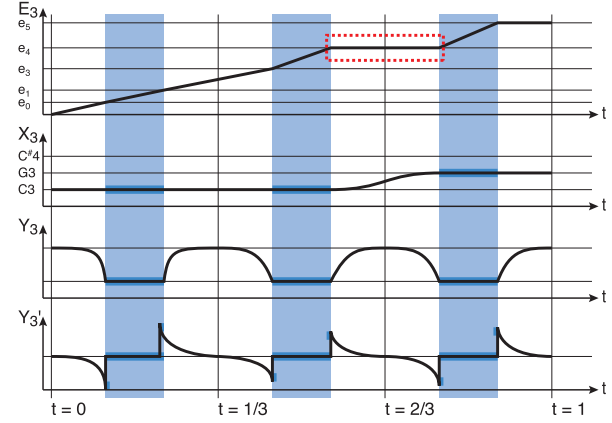


Figure 4. Symbolic gesture curve for finger 3, with curve parameter t running from 0 to 1 on the horizontal axis.

been solved by parameterising onset time for each symbolic finger, i.e., onset time E becomes a function of the curve parameter t : During the transition between the two events, the position coordinate increase, while onset time remains constant. Thus, in a symbolic gesture curve, fingers are allowed to move at ‘infinite’ speed.

For the construction of the curve, each finger has to be handled separately. First, the curve parameter t is divided by the number of events for each finger. Then, each event is divided into three intervals, one for the transition before the event takes place, one for the event itself, and one for the transition after the event. Finally, cubic interpolation is applied for each subinterval. (The interpolation type is however not part of the intrinsic definition of a symbolic gesture curve.)

Figure 3 shows the symbolic gesture curve for finger 2. Each axis is drawn separately in function of curve parameter t . The semitransparent vertical bar denotes the area where the actual event takes place. As we just have seen, onset time E also exists in the gesture space, but separated for each finger. The remaining instrument parameters H , L , etc., are replaced by pseudo space coordinates X , Y , Z , which define the coordinate system for a virtual keyboard. X_2 is the position on the keyboard and corresponds to pitch, Y_2 is the position above the keyboard and tells whether key is pressed or not, and the derivative Y_2' contains information about the speed at which the key is pressed or released, respectively. This speed corresponds to the loudness of a certain event. Note that the Z position (depth on the keyboard, thus defining a white or a black key) is omitted in the figure. The construction is analogous to the one of X . Figure 4 shows the symbolic gesture curve for finger 3. Indicated by the dashed square is the region where onset time E

remains constant since finger 3 has to move from event 3 to event 4 at infinite speed.

3.2 Freezing Symbolic Gesture Curves

While we have just dealt with the construction of symbolic gesture curves, which was denoted by the ‘thawing’ operation in Figure 1, let us add a remark on the reverse process, the ‘freezing’ of symbolic gesture curves. Since the symbolic gesture spaces are similar to the “Note on”, “Note off”, and “Velocity” concepts offered by MIDI, the ‘freezing’ operation in the symbolic domain is easy when compared to the construction of a gesture curve: It is basically the transformation of a MIDI file or a real-time MIDI input, respectively, to an event space, for instance defined by E , H , L , and D (onset time, pitch, loudness, and duration).

Important are possible applications of the ‘freezing’ operation: they provide mechanisms for *recording gestures* from given performances. The most simple, but also the most wide-spread use of such a mechanism is MIDI recording, resulting in a recorded score. Further, the ‘freezing’ operation provides a powerful mapping mechanism from a gestural performance device space (e.g. a gesture tracker attached to a computer) to a musical performance space (e.g. a synthesiser).

4 Construction of Physical Gesture Curves

One goal of the current PerformanceRubette is to calculate physical gesture curves, which then can be represented by avatars or avatar parts in a virtual environment. Unfortunately, we do not know anything about a possible direct transformation $\wp_{Gesture}$ which was given in Figure 1. Thus, instead of defining $\wp_{Gesture}$, we begin with a given symbolic gesture curve, which can for instance be obtained directly from a given MIDI file, ‘freeze’ the curve, which yields events in the score space. From here, \wp_{Score} can be applied, as it has been done in the past (Mazzola, 2002b): \wp_{Score} is defined by performance vector fields, which are numeric results delivered by a number of analyses (e.g. melodic, harmonic, motivic). The process results in physical sound events in the performance space. What remains, is to ‘thaw’ those events, resulting in the physical gesture curve we are looking for. Thus, this section deals with the rather complex ‘thawing’ operation in the physical domain, i.e., the construction of physical gesture curves.

As we shall see now, the concepts of construction of symbolic gesture curves will also be helpful in the physical domain: assume the score in Figure 2 to be a score

that has already been performed, e.g., some MIDI recording. Then, the application of the curve construction algorithms from the former sections yields the symbolic curves given in Figure 3 and Figure 4. However, there are two obvious problems with the constructed curves with respect to the physical domain: first, they might be anatomically incorrect. This is the case for the curve in Figure 4, because it is impossible to play a $C\#4$ with the left index finger while the middle finger still remains at $C3$. Thus, the curve has to be shaped accordingly in order to satisfy this anatomic constraint.

Second, the curve is physically incorrect. As we have seen in the previous section, fingers can move at infinite speed in the symbolic domain. This of course does not hold in the physical domain: a performer needs time to move a finger from one key to another. Furthermore, the curve has been interpolated by cubic splines, but we did not care about the shape of the curve segments – just as long as the required onset times and the onset velocities are satisfied, any interpolation could have been applied. However, in the physical domain, the movements of a mass trough space need to satisfy dynamic laws which requires us to shape the curve accordingly.

4.1 Applying Geometric Constraints

Before we are able to apply geometric constraints, the involved objects need to be specified. In our case of a keyboard performer, the keyboard dimensions need to be defined. Furthermore, a model of the performer’s hands is required. The first issue, the keyboard dimensions, is fairly easy to solve since piano and grand-piano keyboards are well standardised (e.g., DIN8996). In addition, the unshaped physical gesture curves already reside in the space of a virtual keyboard (i.e. the X , Y , and Z coordinates, as defined above). The second issue, the hand model, is much more complicated: hand dimensions and flexibility vary from performer to performer, the numerous joints of the individual fingers, and particularly the special case of the thumb, allow extremely complex movements.

Our approach includes a simplified, joint-based hand-model that consists of 31 geometry parameters (for each finger 1 position vector relative to the hand root, 3 length parameters for each segment; and a special rotation angle for the thumb), and of 25 state parameters (5 for the absolute hand position and orientation, and 4 angles for each finger). In addition, absolute maximum and minimum ranges of the state parameters are defined. These ranges are defined with an eye on possible *performable* positions (with respect to piano performance);

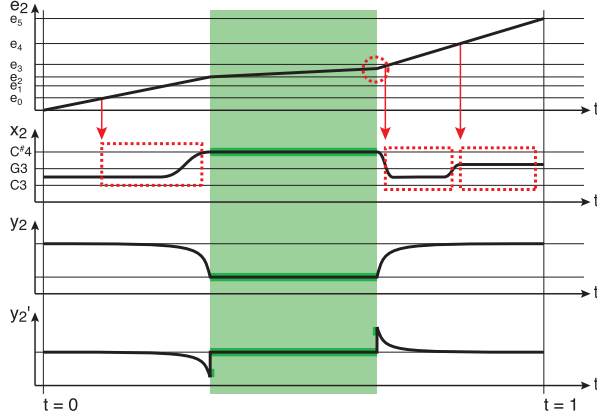


Figure 5. Physical gesture curve for finger 2, with curve parameter t running from 0 to 1 on the horizontal axis.

this allows the elimination of many pathological cases from the beginning.

Now, essential for the application of constraints is the availability of a *decision function*, which tells whether the position of a specific finger with respect to the position of a number of other fingers is possible or not. The decision function basically compares the (Euclidean) distance between the involved fingers and compares it to the allowed distances provided by the specific hand model. By looking again at the example score in Figure 2, one can see that there are regions where fingers are required to be at a certain position, namely, the regions where one (or a number) of the keys has to be pressed and kept down for the duration of the note. Our method uses these regions to apply the geometry constraints: by iterating through the gesture curve, each region is examined. The fingers currently involved in playing an event are examined first. Here, the decision function delivers the answer whether the score in question can be performed at all (in terms of the given fingering). Subsequently, fingers not involved in a note are examined. At this point, we obtain the information if those fingers have to be rearranged, such that the hand remains in an anatomically correct state. The rearrangement is accomplished by replacing the involved interpolation intervals by multiple subintervals.

Figure 5 shows the shaped physical gesture curve of our example. The dashed squares indicate where shaping had to be applied because of failed anatomic constraints. In this case, the curve had to be reshaped three times. The reasons are indicated by the vertical arrows: first, at e_0 finger 3 had to play $C3$, which would have been impossible with finger 2 residing at the position of $C\#4$. Second, at e_3 finger 3 had to play $C3$ again, thus

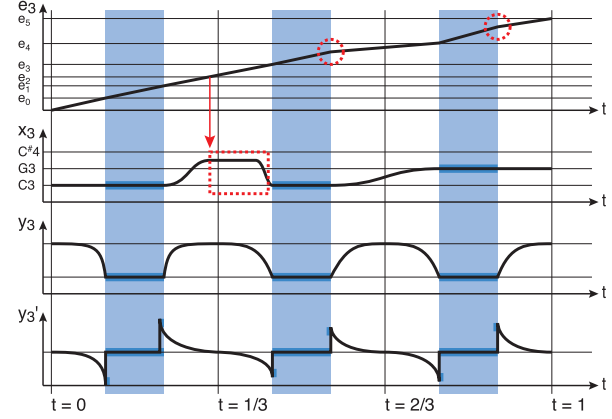


Figure 6. Physical gesture curve for finger 3, with curve parameter t running from 0 to 1 on the horizontal axis.

requiring finger 2 to move back after playing $C\#4$. Finally, at e_4 finger 2 had to be moved above $G3$ to avoid crossing with finger 3, which had to play $G3$.

Although not shown in the example, the method does also work for finger transitions, which occur at all times when playing the piano. Here, special attention has to be paid on replacing interpolated intervals in a way such that the finger tip positions remain in an anatomically consistent state.

Another issue is the question how the curves can be kept anatomically correct when applying further constraints, such as physical model-based shaping shown in the following section. The problem can be attacked by iterating through the individual shaping steps: for example, after applying physical model-based shaping, the geometric constraints are checked once more, and if they fail, the curve has to be rearranged even further (if possible, otherwise, the score has to be considered as not performable), and the remaining steps have to be repeated.

4.2 Physical Model-based Shaping

In order to cope with the physical constraints of finger movements, we have—besides the geometric constraints from anatomy—implemented conditions, which reflect human force limitations when acting upon finger masses. These conditions boil down to the control of zeros of polynomial functions, which is classically provided by Sturm’s theorem (Waerden, 1966, §79).

In order to make the situation conceivable, we want to calculate a simple example, namely the thawing of a curve $G : [0, 1] \rightarrow \mathbb{R}^2$ which describes the change of pitch without intermission, to be performed by determined finger. The curve $G(t) = (e_G(t), q_G)$ has two

components: the curve e_G , measuring physical time, and the pitch curve q_G , measuring physical pitch, as represented by the horizontal distance between the keys of a keyboard. The frozen curve G , as it is shown in Figure 7, draws the change from pitch $q_1 = 0$ to pitch $q_2 = 5$ (think of a fourth leap from C to F), starting at time $e_0 = 0$ for parameter $t = 0$, jumping to q_2 at time $e_1 = 1$ for $t = t_1$, arriving at pitch q_2 at the same physical time e_1 (!) for parameter $t = t_2$, and ending the performance at $e_2 = 2$ for $t = 1$.

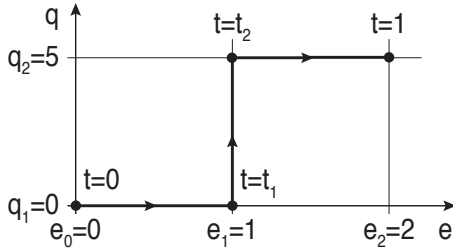


Figure 7. Frozen gesture curve with changing pitch from $q_1 = 0$ to $q_2 = 5$.

The thawing deformation is a new curve $g : [0, 1] \rightarrow \mathbb{R}^2$ with $g(t) = (e_g, q_g)$, which complies with the Newton inequality

$$m \frac{d^2 q}{de^2} < K,$$

where m is the finger's mass, and where K is an upper limit given by the physiological constraints of the performer. The thawed curve g is shown in Figure 8. Evidently, the finger cannot stay fixed on pitch $q_1 = 0$, but has to jump off this position at a physical time $\mu(e_1 - e_0)$, $0 < \mu < 1$, after the start. The main point of the thawing calculations is the position, where the jump begins until its ending on pitch $q_2 = 5$. Denote this curve by $\gamma(x) = (e(x), q(x))$ and suppose the parameter x ranges from $x = 0$ to $x = 1$, so we have two

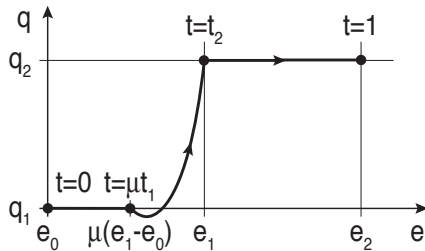


Figure 8. Thawed gesture curve with changing pitch from $q_1 = 0$ to $q_2 = 5$.

boundary conditions $q(0) = q_1 = 0$, $q(1) = q_2 = 5$, $e(0) = \mu(e_1 - e_0) = \mu$, and $e(1) = e_1 = 1$. Then the Newton inequality becomes

$$\frac{d^2 q}{dx^2} \cdot \frac{de}{dx} - \frac{dq}{dx} \cdot \frac{d^2 e}{dx^2} < \left(\frac{de}{dx} \right)^3 \frac{K}{m}, \quad (1)$$

which means that this inequality must hold for all $x \in [0, 1]$. Following a general cubic spline procedure, we model our curves $e(x), q(x)$ by cubic polynomials:

$$\begin{aligned} e(x) &= a_e x^3 + b_e x^2 + c_e x + d_e \\ q(x) &= a_q x^3 + b_q x^2 + c_q x + d_q, \end{aligned}$$

where the boundary conditions were given above. After a normalisation of m, K to yield $K/m = 1$, inequality 1 reads as follows:

$$\begin{aligned} P(x) &= -2b_q c_e + c_e^3 + 2b_e c_q + \\ &(-6a_q c_e + 6b_e c_e^2 + 6a_e c_q)x + \\ &(-6a_q b_e + 6a_e b_q + 12b_e^2 c_e + 9a_e c_e^2)x^2 + \\ &(8b_e^3 + 36a_e b_e c_e)x^3 + \\ &(36a_e b_e^2 + 27a_e^2 c_e)x^4 + \\ &54a_e^2 b_e x^5 + 27a_e^3 x^6 > 0. \end{aligned}$$

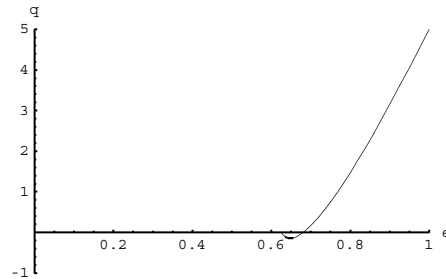


Figure 9. Plot of thawed curve from Figure 8

The Sturm theorem guarantees that no zero of polynomial $P(x)$ occurs in the interval $[0, 1]$ if $P(0), P(1) > 0$, and if the associated Sturm chains $(P(0), P'(0), \dots)$, $(P(1), P'(1), \dots)$ have the same number of signature changes. Recall that the Sturm chains are successive Euclidean algorithms, starting with the division with remainder by the derivative P' , i.e., $P(x) = A(x)P'(x) + B(x)$. These latter calculations are made in advance, the results are then implemented in our code.

The Sturm criterion amounts to the fulfilment of a number of polynomial inequalities $S_i > 0, i =$

$1, \dots, N$, where the polynomials S_i are functions of the curve coefficients $a_e, \dots, d_e, a_q, \dots, d_q$ and the “jumping” coefficient μ , the latter being present from the boundary conditions. One solution to our problem is found by common algorithms and yields $e(x) = 5/8 + x/8 + x^2/4$, $q(x) = -2x + 7x^2$ with $\mu = 5/8$. Figure 9 shows a calculated plot of the curve, meaning that the finger first lowers its pitch position and then leaps to the target pitch.

The dashed circles in Figure 5 and Figure 6 show locations where the anticipated leaps from the precedent keys take place.

5 Results

The PerformanceRubette implements a framework for the realisation of gesture spaces. The framework contains models for symbolic and physical gesture curves of piano-like instruments and supports the “freezing” and “thawing” operations as described in the previous sections. The constraint-based shaping algorithms described in the previous sections have been implemented. Furthermore, the software can read and convert MIDI files to a symbolic gesture curve (under the condition that appropriate fingering information is provided by the user, since it is not contained in a MIDI file), or a MIDI input stream can be processed directly for real-time experiments and applications with MIDI instruments.

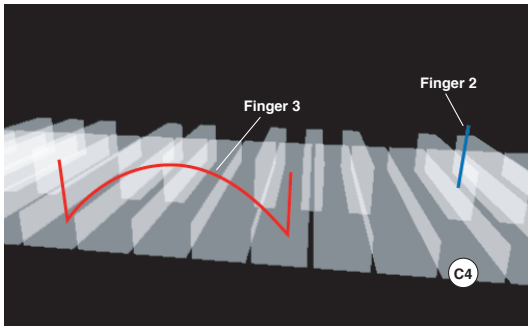


Figure 10. The symbolic gesture curve corresponding to the score from Figure 2.

As a side-effect, we have implemented a generic framework for handling parametric curves of arbitrary dimension. The framework, a number of Java classes, can automatically handle interpolated intervals, derivative and similar operations on coordinate axes. In addition, it is possible to iterate through parameter intervals at an arbitrary resolution.

One of our target applications is the generation of geometry data that can be used to generate interpolated gesture curves ready to be fed into an animation system. Figure 10 shows the constructed symbolic gesture curve from the example score of the earlier sections. The figure does not contain time information, but rather shows the path the two fingers would follow over the whole duration of the score. Figure 11 shows the physical gesture curve, which has been shaped according to the given global constraints. The two examples have been displayed with *Soundium* (Schubiger-Banz and Müller, 2003), a real-time multimedia framework, which was in part developed at the MML.

Another application has been the real-time mapping of a MIDI recording to projected 3D video projections in a live performance: the MIDI stream of a midi grand-piano was used to construct a specialised symbolic gesture curve. In particular, the gesture space differed in the ones from the earlier sections in that it did not contain explicit fingering information (also due to the reason that a MIDI stream does not contain fingering information). Instead, it contained a particular parameter space that could directly be used by the graphics system, such as colour spaces, animation curves, transformations, and 2D video effect parameters. The experiment, see Figure 12, shows that gesture curves can provide a valuable mechanism for bridging between performance interfaces and music and even to other disciplines.

The new version of the PerformanceRubette will be available for download as part of RUBATO (see <http://www.rubato.org>).

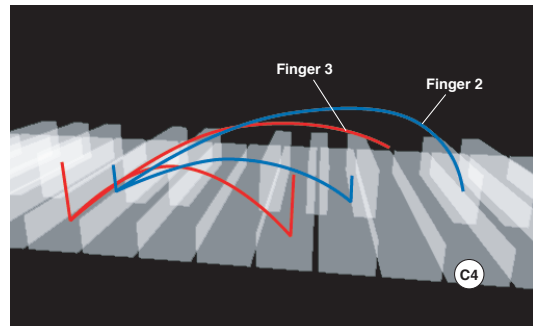


Figure 11. A symbolic gesture curve of a chromatic scale starting at C4, played with the left index finger (finger 2).

6 Conclusions

We have presented the basic concepts of symbolic gesture spaces and curves as an extension to the exist-



Figure 12. Using gesture curves in conjunction with live performance interfaces.

ing score–performance model in performance theory. It was shown how gesture curves can be constructed and shaped in the physical domain, starting out with symbolic gesture curves. The resulting curves can be used for the animation of avatars or avatar parts, or they can be used in conjunction with complex instrument parameters spaces for sound synthesis based on gestural parameters. In addition, the freezing mechanisms support novel gestural performance interfaces.

Current work is mainly focused on the constraint-based shaping in terms of musical constraints. Furthermore the implementation is complemented with inverse kinematics methods in order to be able to calculate the positions not only of the finger tips but also of the inner finger segments and the palm. Of course, the implementation of a complete, realistic hand model imposes a major challenge but will also serve other research directions in computer graphics and animation.

Finally, we hope that our works shortens the path to model instruments other than keyboards. While it should be relatively easy to integrate string instruments, the case of wind instruments is more complex since they will also include facial expression. As an ultimate step we see a unified symbolic gesture space, which can be used for any conceivable instrument, and a direct transformation to the corresponding physical space.

7 Acknowledgements

This work and the RUBATO project are in part supported by SNSF grant 21-59417.99. We thank Peter Stucki for constantly supporting our work.

References

- N. Badler, C. Phillips, and B. Webber. *Simulating Humans: Computer Graphics, Animation, and Control*. Oxford University Press, 1993.
- J. Cassell, H. H. Vilhjalmsson, and T. Bickmore. Beat: The behavior expression animation toolkit. In *Proceedings of ACM SIGGRAPH 2001*, pages 477–486, 2001.
- P. Faloutsos, M. van de Panne, and D. Terzopoulos. Composable controllers for physics-based character animation. In *Proceedings of ACM SIGGRAPH 2001*, pages 251–260, 2001.
- G. A. Henrotte. Music and gesture: A semiotic inquiry. *A J Semiotics*, 9(4):103–114, 1992.
- G. Mazzola. Semiotic aspects of music. In Posner et al., editor, *Semiotics Handbook*, volume III. de Gruyter, W., Berlin, 1999.
- G. Mazzola. Structures mathématiques dans l’interprétation et l’improvisation. In *Séminaire Mamux*, Paris, 2002a. IRCAM Centre Pompidou.
- G. Mazzola. *The Topos of Music*. Birkhäuser, Basel, 2002b.
- G. Mazzola and O. Zahorka. The rubato performance workstation on nextstep. In *Proceedings of the 1994 International Computer Music Conference*, San Francisco, 1994.
- S. Müller. Computer-aided musical performance with the Distributed RUBATO environment. In G. Johannsen and G. de Poli, editors, *Human Supervision and Control in Engineering and Music*, volume 31(3), pages 233–238. Special Issue. J New Music Research, 2002.
- S. Müller. Parametric gesture curves: A model for gestural performance. In G. Mazzola, T. Noll, and T. Weyde, editors, *Proceedings of the 3rd International Seminar on Mathematical Music Theory*. EPOS, Osnabrück, 2003. To appear.
- C. Parrish. *The Notation of Medieval Music*. Norton, W. W., New York, 1957.
- S. Schubiger-Banz and S. Müller. Soundium2 - an interactive multimedia playground. In *Proceedings of the 2003 International Computer Music Conference*, San Francisco, 2003.
- B. L. Waerden. *Algebra I*. Springer, Berlin et al., 1966.
- M. M. Wanderley and M. Battier. *Trends in Gestural Control of Music*. IRCAM Centre Pompidou, Paris, 2000.
- M. M. Wanderley and C. Cadoz. Gesture-music. In M. M. Wanderley and M. Battier, editors, *Trends in Gestural Control of Music*. IRCAM Centre Pompidou, 2000.